Metadata Update Scripts at EDC (MUSE)

Royce Reit, Raytheon ITSS Corp.
Bhaskar Ramachandran & Calli Jenkerson, Emergent IT Inc.
ECS-EDAAC, Sioux Falls, SD (April 13, 2001)

MUSE is a C-Shell script with a number of built-in menus that allow querying and identifying granules before performing QA updates to either the Science or the Operational flags and their Explanations in the metadata tables. These updates are performed via SQL commands directly interacting with the Science Data Server database tables. The changes to the QA flags are accomplished one at a time, but the main merit of using this script is that it does not lock the database tables which used to happen with the previous QA Metadata Update Tool (MUT), thus creating serious contention with the other routine ECS processes.

MUSE consists of the following five files:

MUSE
MUSE_Rc
MUSE_QUERY.sql
MUSE_UPDATE.sql
MUSE_QFCHECK.csh

MUSE: This is a C-shell script that contains all of the menus and the major portion of the code for the MUSE program.

MUSE_Rc: File that defines setup environment variables & DAAC-specific tunable parameters.

MUSE_QUERY.sql: SQL script that queries the DsMdGranules tables in the SDSRV database tables in order to identify and filter a list of granules matching the values defined by the user. The results are written to MUSE_QUERY.out, which is used for performing subsequent procedures.

MUSE_UPDATE.sql: SQL script that updates the DsMdMeasuredParameter table based on the GranuleId(s) and ParameterName(s) that are passed to it. The field/value pairs that are updated are set in the MUSE QUERY MENU based on user input.

MUSE_QFCHECK.csh: This is a C-shell script that edits the value entered for the Quality Flag. The script only allows specific Quality Flag values (Ex: Passed, Failed, etc.). This edit is invoked both when the user enters data and when the stored queries are loaded.

To install MUSE copy the MUSE program, MUSE_QUERY.sql, MUSE_UPDATE.sql, and MUSE_QFCHECK.csh files into a directory with restricted access. Users should not have write access to this directory. For example: The **/tools/common/OPS** directory is used at EDC for storing all DUE scripts.

To run the MUSE program the User should create a directory called MUSE and copy the MUSE_Rc file into this directory. The User may run the program from any directory they choose as long as it has access to the production database. The User should then edit the MUSE_Rc file to set their site-specific information. Once the MUSE_Rc file is setup, the User invokes the MUSE program by calling it from the directory where they installed the MUSE_Rc file. For example: Suppose the MUSE program and files are installed in **/tools/common/OPS** and the user has setup a directory called **/home/user/MUSE**. From the command line in the /home/user/MUSE directory the User would type **/tools/common/OPS/MUSE** and press enter.

NOTE: There is an edit in the MUSE program, which prohibits running the program in the same location that it is installed. This was a requirement for the DUE process.

Once you start up MUSE, it requires a SYBASE login and password to get in. This login and password must have **update** privileges in the SDSRV database. The Main menu has the following sub-menu options:

MUSE MAIN MENU

1. **Load Saved Query Values**
2. **Set SYBASE Values**
3. **Set Query Values**
4. **Run Query**
5. **Quit**

The **Set SYBASE Values** sub-menu has the following options:

MUSE SYBASE MENU

1. **Set Server Name**	(Sybase server for the SDSRV database)
2. **Set Database Name**	(SDSRV database ex: EcDsScienceDataServer1)
3. **Set Login / Password**	(Sybase login & password with update privileges)
4. **Set Working Directory**	(Directory location for writing user output files Ex: /home/bhaskar/MUSE/work)
5. **Set Quality File Directory**	(Directory location for the quality explanation text files Ex: /home/bhaskar/MUSE/Quality_Files)
6. **Return to Main Menu**	

The Query values are set via the **Set Query Values** sub-menu, which contains the following options:

MUSE QUERY MENU

1. **Set ShortName** (ESDT Short Name)
2. **Set VersionID** (ESDT Version ID)
3. **Set Quality Flag Value** (Qualifier: Passed/Failed ….)
4. **Set Quality Flag Explanation File** (Name of the text file in the Quality File Directory which contains the quality explanation narrative Ex: Expln.txt)
5. **Set Quality Flag Type** (S = Science, O = Operational)
6. **Set Acquisition Range** (If entered, the query can be restricted by a beginning and ending acquisition date range Ex: Jan 1 2000 12:01AM - Dec 31 2000 11:59PM)
7. **Set LocalGranuleId** (If entered, the query can be restricted to a single LGID or multiple LGIDs using the "%" wild card. The default is ALL LGIDs)
8. **Restore Default Values (5-7)** (Restores default values to options 5 through 7)
9. **Return to Main Menu**

When the query is run via **Run Query** on the Main Menu, the following results menu is displayed:

MUSE RESULTS MENU

1. **Save Query Parameters** (Allows the user to save the user-defined query parameters which can be loaded and run at a later time)
2. **Review Query Results On-line** (Allows the use to view the QA Flag and Explanation values before update)
3. **Run the Update** (Updates the SDSRV database for each granuleId/ParameterName combination returned in the query)
4. **Review Update Results On-line** (Allows the user to view the QA Flag and Explanation values after update)
5. **Return to Main Menu**

The saved Query Parameters can be loaded by the user from the Main Menu. This eliminates the need to re-enter all the same query values each time you start MUSE

We have used these scripts to run tests of up to 1000 granules, which took about 2.3 seconds per granule (total time of under 40 minutes) at a time when all the other routine processes (ingest, search, and distribution etc) were simultaneously taking place.

Additional Functionality to Add to MUSE

?? Supporting multiple LGIDs as input (upto 50,000 LGIDs)

?? Functionality to handle multiple Measured Parameters within the same granule

?? Error handling capabilities

?? Log writing and maintenance capabilities

?? Convert the MUSE_QUERY.sql and MUSE_UPDATE.sql files to Stored Procedures to enhance the efficiency of MUSE

?? Adding a Graphical User Interface